REMARKS

Claims 1-5, 7-9, 18-21 and 24 stand rejected as allegedly being anticipated under 35 USC § 102(e) by US Patent No. 6,230,274 (hereafter Stevens).

Claims 6, 10-17, 22-23 and 25 stand rejected as allegedly being unpatentable over Stevens in view of US Patent No. 6,272,642 (hereafter Pole).

Applicant has amended the specification to remove the hyperlink and to improve trademark citations.

In order to set forth a prima facie case of anticipation, a reference must show each and every element of the claims. Applicant respectfully submits that Stevens does not show each and every element of that which applicant now claims.

Stevens discusses resuming a memory subsystem from a low power state such as a suspend to RAM (STR) state. A suspend to RAM state is materially different than many other low power states in that power is removed from enough components that system state is deemed to be corrupted by the process of suspending to RAM. As a result of such suspension, as described by Stevens, any data which the user wishes to preserve must be stored to some non-volatile medium or some medium to which power is retained.

> A suspend-to-RAM (STR) state is a common state used in power management applications. Typically, when this low power state is entered, processing activity ceases, and certain values are stored in memory, preserving them for when processing resumes at a later point in time. For example, the ACPI S3 sleeping state is a state where all system context is lost except system memory. Processor and memory controller context (i.e., register and internal memory values) are lost in this state. Additionally, other power management techniques may include similar states in which the register and/or memory values in a chipset or memory controller are lost.
>
> Stevens, Col. 1, lns. 32-44.

Thus, to resume from an STR state, a system is effectively restarted as if it was being rebooted. However, some non-volatile indicator checked in the boot process directs the flow of instructions to resume the suspended state rather than to reboot the machine from scratch. See also US Patent 6,438,668, cited in the Office Action, particularly block 651 in Figure 6b and Col. 9, lines 25-36, et seq. The BIOS plays a key role because it is a separate memory that can be accessed without initialization such as that performed for the Stevens memory subsystem. Thus, the BIOS begins the boot process and is able to perform operations before the memory subsystem is brought to an operable state. Program instructions from other regions of memory stored in the memory subsystem simply could not be executed because those other regions of memory cannot be accessed until the memory subsystem is operational.

Stevens explains that it is the BIOS that powers up the memory controller and loads such information so that the memory subsystem can be accessed.

> After entering the STR state, an event which causes the system to exit STR may be sensed as indicated in block 1010. Accordingly, the BIOS powers up the MCH and other system components. The configuration registers of the MCH may be automatically reset to a default value in this process.
>
> Accordingly, to again access memory devices on the memory channel, at least some of the configuration register values are needed. The BIOS may cause the ICH 505 to access the battery backed-up memory 590 and restore the registers listed below (saved in block 646 of Figure 6).
>
> - MCH GAR registers (040-047h)
> - MCH RDT (050h)
> - MCH DRAMC (051h)
> - MCH PGPOL (052h)
> - MCH RPMR (053h)
> - MCH GBA registers (060-6Fh)
> - MCH Configuration Registers MCHCFG (0BE-BFh)
>
> After restoring values to these registers, the MCH can once again access items stored in memory when the STR state was entered, including such items as the processor context if saved. The memory devices perform self-refresh in the STR state so other data is not lost.
>
> Stevens, Col. 19, lns. 17-40 (emphasis added).

Thus, when an STR state is exited, the BIOS is activated. According to Stevens, the BIOS can be configured to restore variables that allow the memory subsystem to be accessed. In this context, Stevens provides an optimization whereby the full re-initialization of the memory subsystem is not performed. In particular, channel levelizing and other initialization activities may be omitted in a resume from RAM sequence according to Stevens.

Thus, applicant respectfully submits that Stevens employs a BIOS-based restart, which involves having the processor execute BIOS instructions. This BIOS based re-start differs fundamentally from that which is now claimed by applicant. Applicant claims in claim 1:

> 1. An apparatus comprising:
>
> a processor;
>
> an operating system to control a plurality of power management states, one of said power management states being a low latency low power state;
>
> a memory subsystem that requires initialization commands to exit a memory low power state;
>
> control logic to detect exiting of said low latency low power state and to responsively generate a plurality of initialization commands to remove said memory subsystem from said memory low power state prior to allowing execution of the processor to resume.

Applicant claims that some control logic is provided in the apparatus. That control logic performs some acts before allowing execution of the processor to resume from the low power state. As applicant explained in applicant's background section, some low power states, other than STR, do not vector back to the BIOS upon exit. In fact, some low power states resume directly to the operating system (or may resume to other software that is stored in the memory subsystem). The problem with resuming to

software that is stored in the memory subsystem is that in memory subsystems that would require some initialization prior to access, it would not be possible to resume to software stored in the memory subsystem because that subsystem would not be initialized.

In contrast, in the system of Stevens, there is no need to have such control logic because the system begins the resume process by running BIOS, which can be executed without such initialization. Therefore, in Steven's system there is no logic that prevents the processor from beginning to execute instructions as the STR state is exited. Rather, when the STR state of Stevens is exited, the BIOS begins to execute. Thus, Stevens teaches away from applicant's solution claimed in claim 1 because Stevens advocates the execution of instructions by the processor. Stevens does not describe preventing the processor from executing BIOS because the BIOS is contained in a memory that does not require extensive initialization, but rather is readily accessed.

One use of the apparatus claimed in claim 1 may be to resume from a low latency low power state in which BIOS is not activated. This creates a vexing problem. If the BIOS is not activated, and initialization is required to access the memory subsystem, how can the processor resume operations normally by executing instructions? The problem is that the memory subsystem will not be accessible. To overcome this problem, applicant's control logic prevents the processor from allowing memory-access-generating execution to resume immediately. Instead of allowing execution of the processor to proceed to the point where memory accesses would be generated, the control logic holds off that execution until it can generate a plurality of initialization commands to remove the memory subsystem from its non-initialized and inaccessible state.

In contrast, the system described by Stevens allows processor execution to occur because that execution starts from the BIOS, which can itself deal with the un-initialized memory. In contrast, applicant's apparatus claimed in claim 1 allows resumption of a complex, initialization-needing memory subsystem from a state which does not revert to BIOS code. Applicant believes that this difference is clearly reflected in claim 1 and justifies allowance of claim 1 over the Stevens reference.

The Office Action alleges that the control logic is shown by Stevens at Col. 21, line 11. However, the claims in Column 21 merely recite that initialization operations are performed. As noted above, Stevens teaches performing such initialization operations via BIOS. Thus, at least because BIOS is executed by the processor, applicant's claim 1, which recites that the plurality of initialization commands are generated prior to allowing execution of the processor to resume, can not be read on the system of Stevens.

With respect to claims 2, the Office Action argues that Stevens teaches the resumption from the ACPI S1 state. This is simply not true. Stevens discusses resumption from the STR state, such as an S3 ACPI state. The mere mention of ACPI states does not mean that Stevens teaches one to resume from each and every state.

Similarly, with respect to claim 3, the Office Action indicates that it would simply be obvious to resume without using BIOS routines. However, it is not clear from Stevens how one would overcome the problem of not being able to execute from the memory subsystem (due to lack of initialization of the memory subsystem), unless BIOS is used for initialization. Therefore, applicant maintains that it would not be obvious to resume such a memory subsystem without using BIOS.

With respect to claims 5 and 24, the Office Action argues that the deassertion of a stop clock signal after the initialization commands have been performed is taught by Stevens. However, this is not true. Stevens must deassert the stopclock signal to the processor before sending the initialization commands. Indeed, the initialization commands of Stevens are performed by the BIOS. Therefore, the processor is executing instructions in order for the BIOS to be operating, and the clock of the processor is not stopped at this point in time. Therefore, it is incorrect to say that Stevens teaches deasserting the stopclock signal after the initialization commands.

With respect to claim 10, applicant notes that memory system resume logic is provided to hold off processor execution until some initialization can be performed. In particular, claim 10 states that the memory system resume logic is to "sequence through a plurality of initialization commands prior to generating a signal to cause a processor to exit a low power state". This differs from what is taught by Stevens. In Stevens' system, the processor does exit the low power state and begin executing the BIOS before the memory subsystem values are restored. Therefore, the system of Stevens does not do this type of initialization before signaling the processor to exit the low power state. The processor actually is an active participant in executing the instructions that cause the initialization to occur. In contrast, in one disclosed embodiment of applicant's invention, separate logic separate from the logic that normally executes instruction sequences is used to perform the initialization commands.

> The embodiment of Figure 1 utilizes logic in both the ICH 180 and the memory interface 110 to exit from a low latency low power state and to appropriately initialize a memory subsystem 160 so that execution may be resumed. Accordingly, the memory subsystem 160, the operating system 170 and the processor 140 need not be involved.
>
> Applicant's Disclosure, p. 7.

This approach is quite different from that of Stevens, whereby the processor itself executes instructions of the BIOS to restore memory subsystem values.

With respect to claim 18, the system of Stevens invokes the processor's instruction execution resources in the process of resuming from the suspend to RAM state. As previously discussed, the suspend to RAM state invokes the BIOS and largely restarts the computer, restoring the previous state. The BIOS is executed by the processor. Therefore, the processor's instruction execution resources are fully and directly invoked. Thus, initialization is performed prior to resumption to software code that is stored in memory that would be inaccessible but for the initialization.

Conclusion

Applicant has given at least one reason justifying patentability of all claims, and has not attempted to point out the numerous ways to justify patentability of all the different claims.

As to any remaining combinations formed by dependent claims and not specifically addressed, applicant does not concede that they are obvious or anticipated. Rather, rejections of these claims are overcome since at least the base combination is not anticipated nor obvious in view of the prior art. Consequently, applicant submits that there also can be no motivation shown in the art to form the additionally limited combination claimed in such dependent claims since the prior art does not anticipate or make obvious the base combination.

Amendments which are not specifically discussed with respect to overcoming a particular art objection have not been made in order to overcome the prior art.

Applicants submit that all claims now pending are in condition for allowance. Such action is earnestly solicited at the earliest possible date. If there is a deficiency in fees, please charge our Deposit Acct. No. 02-2666.

Respectfully submitted,

Date: 12/10/2002

Jeffrey S. Draeger, Reg. No. 41,000

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8598

APPENDIX A

VERSION OF CLAIMS WITH

MARKINGS TO SHOW CHANGES MADE

18. (Amended) A method comprising:

detecting an event to cause an exit from a low latency low power state;

initializing a memory subsystem transparently to [an operating system]execution resources of a processor that is starting operations in response to the exit from the low latency low power state;

exiting the low latency low power state.

23. (Amended) The method of claim 22, after initializing, further comprising:

returning [a]an initialization complete message to the I/O control hub;

deasserting a stop clock signal.

FIRST CLASS CERTIFICATE OF MAILING
(37 C.F.R. § 1.8(a))

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage in an envelope addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231

on December 10, 2002.
Date of Deposit

Conny Willesen
Name of Person Mailing Correspondence

Conny Willesen — 12-10-02
Signature — Date